Eight Tips to Start With Python

Tarek Ziade from a blog post dated September 24th 2007 on his blog http://tarekziade.wordpress.com/

A friend of mine is starting Python. I tried to sum up some tips for him, that may be useful to others. Don't hesitate to comment it if you think something important is missing.

1. **Get the best online documentation**.

   There are a few online documentation you must read:

   - the official tutorial, that gives you a quite complete overview of Python;
   - the standard library module index. You can download it to simplify the search through greps. This is the documentation you get through the *help* command in the prompt.
   - Active State's Python Cookbook. There are thousands of code snippets that are created, ranked, categorized and commented by developers.
   - Dive Into Python online book, that makes you discover Python features through well thought examples.

2. **Read PyCon, EuroPython and Pycon UK wrapups and slides**.

   They are the three main Python events, and a lot of things are happening there. You'll learn a lot by reading the talks slides. If you can go there, it's even better: sprints, bird of feathers and lighting talks are organized. To convince your boss to send you there, you could make a talk proposal "My first steps in Python" ;)

3. **Suscribe to the right feeds**.

   - The mainstream is Planet Python. It gathers most of the blogs out there, so it is the best place to start.
   - Pythonware's Daily Python URL. Human-filtered feed. It used to provide several dozains of links per week, but it seems to have slowed down, and provides a few links a week now. I think it's better this way.

4. **Learn and use the rising standards**.

   There are a few libraries that have a deep impact on the way people write and distribute their work:

   - setuptools: helpers to build and distribute your code eggs. A public repository à la Perl's CPAN called Cheeseshop is wired with this library so people can distribute their code there. It's one of the major innovation of last years in Python world in my opinion.
   - sqlalchemy: The ORM that is now used by the majority of Python frameworks. Its flexibility is impressive. I think there is no equivalent tool in any other language (please let me know if there is);
   - Python paster. This tool allows you to create templates that can be used to generate skeletons for your code. It is used by many web frameworks to provide people a simple way to generate a standardized boiler-plate code canvas when they start up something. This is done in Java for quite a long time (you cannot do without it in Java, otherwise it would take you years to write any program ;)), and tools like PyDev and Eclipse would provide the canvas to do similar things. But the paster is independant from any IDE;
   - reStructuredText: learn how to use it. It's our LaTeX. Your code documentation should use it.

   I am sure they are other tools out of my domain of expertise that are major. I am thinking of libraries in the scientific world for example.

Anyone can help me on this topic to complete this tip ?

5. **Ask for help**. The three places you can get some help are:
   - the mailing list
   - the irc channel *#python* on freenode.
   - the tutor mailing list. Mihai Campean says: *"This is a list specifically for those new to Python and those interested in helping people learn the language, and the atmosphere is very friendly. It's probably a better place to start than python-list, in my opinion"*

   There are some talented guys that dedicate their free time to help newcomers.

6. **Try to adapt your successfull code patterns**.

   When I started Python, I tried to adapt what I used to do with the tool I mastered then (Delphi). Since *There should be one– and preferably only one –obvious way to do it.* (try *import this* in a prompt), that helped me a lot to learn and understand all the subtles of Python on use cases I mastered. The most pleasant thing about it is that you quickly drop all Python books and guide to work with the language, unlike Java for example, where you need to keep many reference books on your desk.

7. **Share on your experience and participate !**.

   A newcomer (yeah! fresh blood!) experience is a highly valuable material for the language advocacy: the discovering state of mind sometime reveals weaknesses or absurdities experienced users don't see anymore. Furthermore, fresh new ideas are often brought by people that comes from other communities. If you feel that something is absurd, unclear or wrong, you should start a thread on the language mailing list. If you have an idea onany kind of enhancement, maybe it worth a Python Enhancement Proposal.

8. **Watch what is being done in Python 3, PyPy and web frameworks**

   Python 3 is the next version of Python, PyPy is Python written in Python. Web frameworks like Django or Zope are large Python codebases. These three *sub-communities* have something in common: they form the R&D of the language. *Zope* for example, has enhanced a lot *setuptools* and *doctest* through a massive feedback. Keeping an eye on them even if you don't use them will make you live and understand what rises in the language.

   PyPy is an amazing project. Even if you don't understand everything (Python in Python ? what the… ;)), seeing one of Armin Ringo talks will give you an instructive high level view of Python. Now for Python 3, even if you cannot read and understand all threads in the dedicated mailing list, keeping an eye of Guido's wrapups and thread subjects will help you to do the jump on P3k, and probably make your Python 2 code look nicer.

Posted by Tarek Ziadé
Filed in python

8 Responses to "Eight tips to start with Python"

1. *Mihai Campean* said:
   September 24th, 2007 at 8:59 am

   Thanks for the useful tips Tarek, I am also starting to learn Python and this kind of information is just what I am looking for. However there is a subject less touched by you in this post, and this is good IDE's for Python. I would really appreciate some recommendations on this subject.

2. *Kalle Svensson* said:
   September 24th, 2007 at 9:08 am

   I'd also recommend the tutor@python.org mailing list (http://www.python.org/mailman/listinfo/tutor). This is a list specifically for those new to Python and those

interested in helping people learn the language, and the atmosphere is very friendly. It's probably a better place to start than python-list, in my opinion.

3. *Michael Easter* said:
September 24th, 2007 at 11:35 pm

   IMHO, the single most important tip is to get one's hands dirty! Either with quick scripts or in the console... Then find a pet problem (preferably known from a previous language) and implement it.

   Also, experiment, play, and be curious. Rediscover the joy of programming.

   Experimentation is always rewarding but _especially_ with Python. It's like holding a strange substance that can be liquid and solid at the same time.

4. *Alan* said:
September 25th, 2007 at 12:44 am

   The book that helped me most was "Dive Into Python".

   More info per page than most, but all explained clearly and cleanly.

   Find it at http://www.diveintopython.org/

5. *Tennessee Leeuwenburg* said:
September 25th, 2007 at 1:19 am

   Hi Tarek,

   I thought your article was really great. Also, thanks to the commenter who mentioned tutor@python.org. I didn't know about this (or forgotten about it) and I got a real jolt of "gee, that's a good idea" when I read your comment.

   If you would be interested in including your article in the next issue of The Python Papers, I'd love to include this post. Contact me at tleeuwenburg@gmail.com if you'd like to follow up...

   Thanks,
   -Tennessee

6. 

Tarek Ziadé said:
September 25th, 2007 at 7:11 am

   @Mihai: Thanks for the tip. I think this is a good starting point here: http://spyced.blogspot.com/2005/09/review-of-6-python-ides.html. I am going to see how I can integrate it. It's not easy because it's a topic on itself 

   @Kalle: Thanks, great to know, I added it.

   @Michael: You are right, and the prompt is really neat for that.

   @Alan: Added, thanks.

   @Tennessee: Thanks ! Sure i'll send you a mail

7. *osKanpur* said:
September 25th, 2007 at 2:12 pm

   Thanks for the Python tips. I am looking for an easy to "get hands dirty" resource for introducing children to Python programming. Is there any resource which you could point to for children ages 10 - 15 learning their first programming language ? Maybe some simple interactive resource which sums up great ways to do maths or

algebra or in fact any typical school problems with Python ? Many Thanks osKanpur

8. *Mihai Campean* said:
   September 28th, 2007 at 10:05 am

   The review of the 6 python IDE's is a good article, thanks for the pointers.